

Creating HTML forms using KompoZer

1

With increase in the use of Internet many activities have become online. We may use a web page to fill information about ourselves or a product. HTML forms are used to help the visitors of the website to input data. It allows for more interactivity and control in data entry. For instance, if you want to open a mail account or register on a website, you need to enter your personal details in a form. This information is used to setup the account for the user. In order to obtain such information on Internet, HTML forms are used. This data is further stored by the application and used to retrieve the details about the users registered on the website.

A form in HTML is a container used to collect different kinds of inputs from the user. HTML forms contains elements like label, checkbox, text input field, radio button, submit button, reset button and many more. These elements are used to enter the data as well as validate the data within the forms. We will create a simple form to enter the data using HTML tags, but before that let us discuss about the elements used to create HTML forms. The elements used are described in the section below :

- Form
- Input
- Textarea
- Select and Option

Form elements

The form element is used to create an HTML form. It acts as a container for all the elements used in the form. The tag `<form>... </form>` is used to implement this element. The example shows the use of the form element :

```
<form action="register.html" method="post">
```

input elements

```
</form>
```

Observe that the form element uses two attributes namely action and method. The action attribute is used to specify where to send the form data when the form is submitted. It takes a filename as value. This file is opened when the user clicks on the submit button after filling the data in the form.

The method attribute specifies the HTTP method to be used when sending the data. It can take

two values; GET and POST. The GET method retrieves the data from the form and sends it to the server by attaching it at the end of the URL. This method allows only a limited amount of information to be sent at a time. In the POST method, the data is sent as a block through the HTTP transaction. The data is included in the body of the request. This method does not have any restrictions on data length. The default value for method attribute is GET.

Input element

The input elements are used to insert various fields like radio button, text box and checkbox in the form. The tag `<input></input>` or `<input>` is used to implement this element. The input tag has different attributes like type, name and value.

The type attribute of the input element specifies the field that is to be created in the form. The name attribute specifies the name to be used for the field in the form. The value attribute specifies the default value of the field in the form. Table 1.1 shows different values of attribute and its usage.

Type	Description	Example
Radio	Creates radio buttons in the form. Any one radio button can be selected at a time from a group of radio buttons. Generally used to select a single item from a given group of items.	<code><INPUT TYPE = "radio" NAME = "var" VALUE = "txt"></code>
Checkbox	Creates checkboxes in the form. Multiple checkboxes can be selected at a time. Generally used to select a multiple items from a given group of items.	<code><INPUT TYPE="checkbox" NAME = "var" VALUE ="txt" ></code>
Text	Creates a text field to enter text in the form. A user can enter any data of his choice in the text field.	<code><INPUT TYPE = "text" NAME = "var" VALUE = "txt" ></code>
Password	Creates a password field in the form. Similar to the text field but the characters are not displayed to the user. Instead the character typed is converted into non readable format.	<code><INPUT TYPE = "password" NAME = "var" ></code>
Submit	Creates a submit button in the form. On clicking the submit button, the values of data entered in the form is submitted to the file specified in the action attribute of the form element.	<code><INPUT TYPE = "submit" VALUE = "label" ></code>
Reset	Creates a reset button in the form. On clicking the reset button, the values of data entered in the form are cleared and set back to default values.	<code><INPUT TYPE = "reset" VALUE = "label" ></code>

Table 1.1: Values of Type attribute used with input tag

Textarea element

The Textarea element allows multi-line text input. The tag `<textarea>...</textarea>` is used to implement this element. It allows entering unlimited number of characters. It can be used to enter comment, report or a long description of product. The size of a textarea element can be specified using rows and cols attributes. The rows attribute is used to set the number of rows of text that will be visible without scrolling up or down. The cols attribute is used to set the number of columns of text that will be visible without scrolling right or left. The following example shows how to insert a textarea in the form.

```
<form method="post" action="comment.html">  
    Input your comments: <br /> <textarea name="comments" rows="4" cols="20">  
        ... Your comments here...  
</textarea>  
</form>
```

Select and Option element

The select element is used to create a drop down list or menu in a form. The option element is used to specify the values that are to be displayed in the menu. The tag `<select>....</select>` is used to create a drop down menu. The tag `<option>...</option>` is used to create the elements within the menu. Following example shows the use of select and option element.

```
<select>  
    <option value="Ahmedabad" >Ahmedabad</option>  
    <option value="Rajkot" >Rajkot</option>  
    <option value="Surat" >Surat </option>  
</select>
```

Let us now create a sample registration form using the elements learned so far. Code listing 1.1 shows the HTML source code used to generate the form. The output of the code is shown in figure 1.1.

```
<HEAD>  
    <TITLE>Registration Form</TITLE>  
</HEAD>  
<BODY bgcolor="lightblue">  
    <h1> <center>Registration Form</center></h1>  
    <FORM name="frmRegistration" action="form.html">  
        <center>  
            <TABLE BORDER="0">  
                <TR>  
                    <TD width="12%">First Name</TD>  
                    <TD width="1%">&nbsp;</TD>
```

```

<TD> <INPUT type="textbox" name="txtFirstName"></TD>
</TR>
<TR>
    <TD>Middle Name</TD>
    <TD>&nbsp;</TD>
    <TD><INPUT type="text box" name="txtMiddleName"></TD>
</TR>
<TR>
    <TD>Last Name</TD>
    <TD>&nbsp;</TD>
    <TD> <INPUT type="text box" name="txtLastName"></TD>
</TR>
<TR>
    <TD>Gender</TD>
    <TD>&nbsp;</TD>
    <TD>
        <INPUT type="radio" name="Gender" value="male" CHECKED>Male
        <INPUT type="radio" name="Gender" value="female" >Female
    </TD>
</TR>
<TR>
    <TD>Hobby</TD>
    <TD>&nbsp;</TD>
    <TD>
        <INPUT type="checkbox" name="chkSinging" value="Sing" CHECKED>Singing
        <INPUT type="checkbox" name="chkDancing" value="Dance">Dancing
        <INPUT type="checkbox" name="chkReading" value="Read">Reading
    </TD>
</TR>
<TR>
    <TD>Address</TD>
    <TD>&nbsp;</TD>
    <TD>
        <Textarea name="txtAddress" rows="5" cols="70">Insert Address Here</Textarea>
    </TD>
</TR>
<TR>
    <TD>City</TD>
    <TD>&nbsp;</TD>
    <TD>

```

```

<Select Name="cmbCity">
<Option >Ahmedabad</Option>
<Option >Baroda</Option>
<Option selected>Rajkot</Option>
<Option >Surat</Option>
</Select>
</TD>
</TR>
<TR>
<TD>&nbsp;</TD>
<TD>&nbsp;</TD>
<TD> <INPUT type="submit" name="cmdSubmit" value="Submit">
<INPUT type="reset" name="cmdReset" value="Reset">
</TD>
</TR>
</TABLE>
</center>
</FORM>
</BODY>
</HTML>

```

Code listing 1.1: HTML code to generate sample registration form

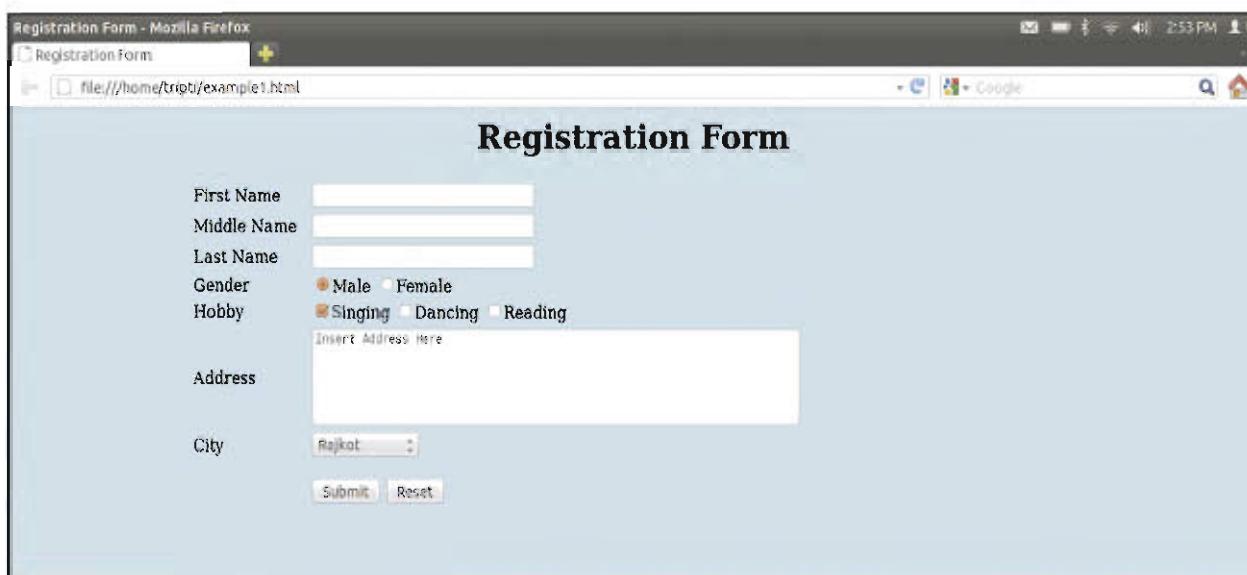


Figure 1.1: Registration Form as displayed in the web browser

Thus, you can see that creating a form using HTML tags is a tedious process. A simpler method is to use an IDE (Integrated Development Environment). An IDE is a software application that provides complete facilities to programmer to develop software. It provides a GUI (Graphical User Interface), text or code editor, a compiler and/or interpreter and a debugger. KompoZer, Eclipse, JBuilder and Netbeans are all examples of some open source IDEs. Let us discuss how to use the KompoZer to create web pages.

Introduction to KompoZer

KompoZer is a free open source web development IDE. It can be downloaded from <http://www.KompoZer.net>. It provides a web page editor which has a simple graphical interface known as WYSIWYG "what you see is what you get". It is a complete web authoring system which integrates web page development and web file management. Creating new web pages using KompoZer is quick and easy. The users can also edit the web pages by using the source code and making changes. KompoZer incorporates a Site Manager which gives rapid access to the files on both local machines and remote servers. Web pages and associated files can be uploaded to a remote server from within KompoZer. It also supports the use of "Styles" through Cascading Style Sheet (CSS). We will learn about CSS in the next chapter.

Before we learn how to create forms using KompoZer, let us first learn the KompoZer interface. Open KompoZer by locating its icon. To view the different toolbars and status bar (if not visible) click on **View → Show/Hide**. Select all the options listed: Composition Toolbar, Format Toolbar1, Format Toolbar2, Edit Mode Toolbar and Status bar. Site Manager and Rulers option should also be checked. Figure 1.2 shows the window that will be displayed after selecting the toolbars.

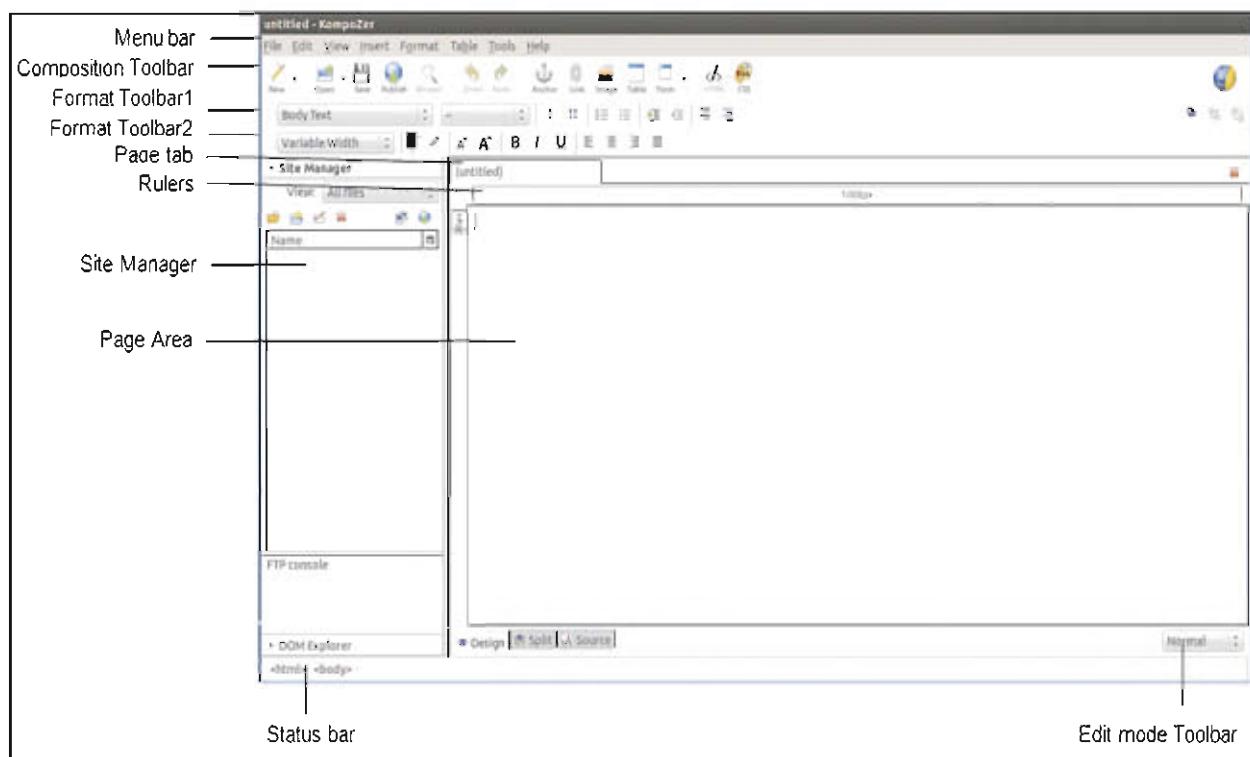


Figure 1.2: KompoZer Interface

Observe that in figure 1.2 you can see the menu bar on the top with the options like File, Edit, View, Insert, Format, Table, Tools and Help. Below the menu bar there are three Toolbars: Composition, Format Toolbar1 and Format Toolbar2.

The composition toolbar is used to create new file, open a file, save a file or publish a web page. The Format toolbar1 and Format toolbar2 are used to format the text, add bullets, numbering and perform similar formatting operations.

In the centre of the window, you can see two panes: Site Manager and blank web page. Site manager is a powerful tool used to navigate within the site or between the sites. You can close the site manager pane by clicking on close button or press F9. The page pane shows a blank untitled web page. The bottom right side of the window shows Edit mode toolbar with three viewing modes: Normal, HTML Tags and Preview. All the three viewing modes provide editing facilities.

The Preview mode offers the page view as seen in a browser. The difference is that in the preview mode the scripts do not run and therefore their effects will not be seen. The links also does not operate in preview mode.

The Normal view is very similar to preview mode. In this mode the table outlines are visible.

The HTML Tags view helps those who are familiar with HTML. A yellow marker is used to indicate the start tag for all elements. Clicking on a marker selects and highlights whole of the element.

The left side of the page pane shows Design, Split and Source tabs. The Design tab is used to design the web page. The Split tab displays the HTML source of the current element. Source tab shows all details of the HTML code. It helps in editing the source code.

On the bottom of the window you can see the status bar. When we click on any item in page, its structure appears in the status bar. If you want to customize any toolbar, right click on the respective toolbar and click on customize toolbar option. You can then customize the toolbar as per your choice.

Create a New File

To create a new file, open KompoZer. In the menu bar click **File → New**. A dialog box as shown in figure 1.3 appears with title "Create a new document or template". Select "Blank document" option from the options that are visible in the dialog box. At the bottom of the dialog box you can see a label "Create in". Select New Tab option from the drop down menu next to it. This allows us to create the page in a new tab. Click on the Create button.

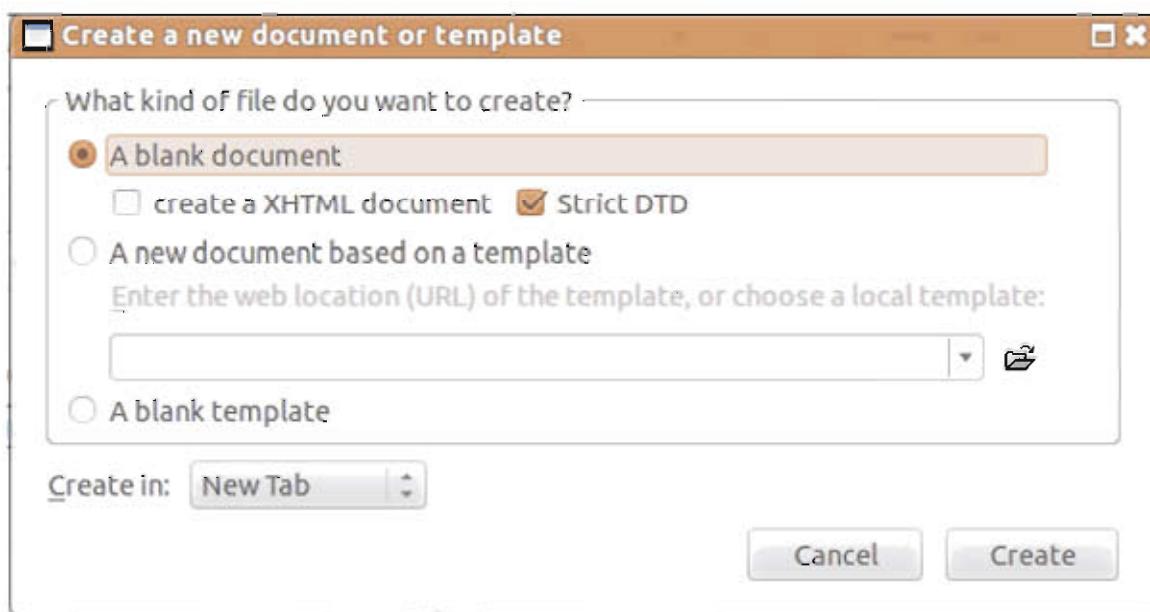


Figure 1.3 : Create a new file

Open an existing File

To open an existing file, click on  icon present on the composition toolbar. Alternatively, you can also click on **File → Open**. If the file has been opened recently, then you can also open the file from **File → Recent Pages**.

Now let us learn to create a forms using KompoZer. We will create a simple form with two input fields: Name and E-mail address and a submit button. Follow the steps given to create the form.

- Open KompoZer. Create a new file.
- From the menu bar, select **Insert → Form → Define Form**. Alternatively, you can also click on  in the composition toolbar. This will open a Form Properties dialog box as shown in figure 1.4. Clicking on More Properties shows added options for the form.

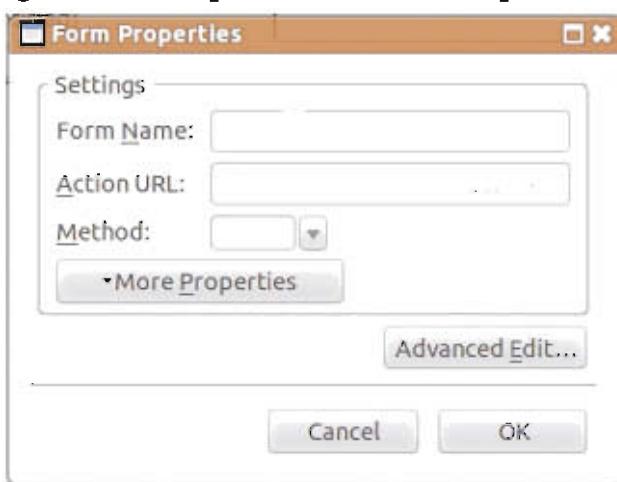


Figure 1.4 : Form properties dialog box

- Enter appropriate name for the form. In the Action URL, enter the file name where you want the form data to be submitted. Select the method POST from the method drop down menu, and click on the OK button. Figure 1.5 shows the details added in the form property dialog box.

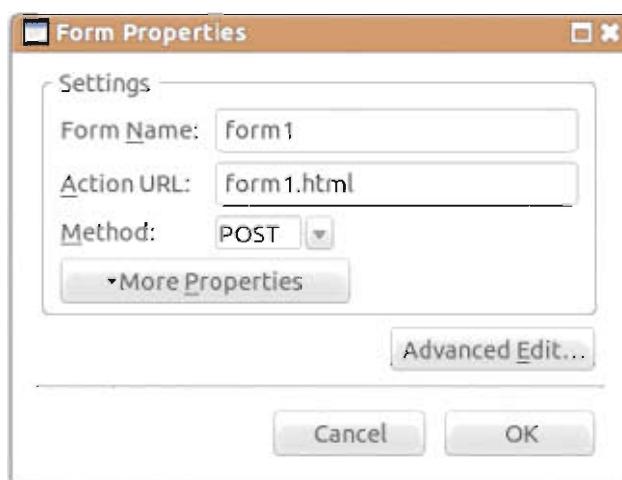


Figure 1.5 : Details entered in Form properties dialog box

- You can see the form inserted with light blue colored outline in the untitled page as shown in figure 1.6. In normal view, the forms are shown surrounded by a dotted blue box. All the form elements like text box, radio button, check box and drop down box will be placed within this box. Press the Enter key a few times to give some space to work on the form.

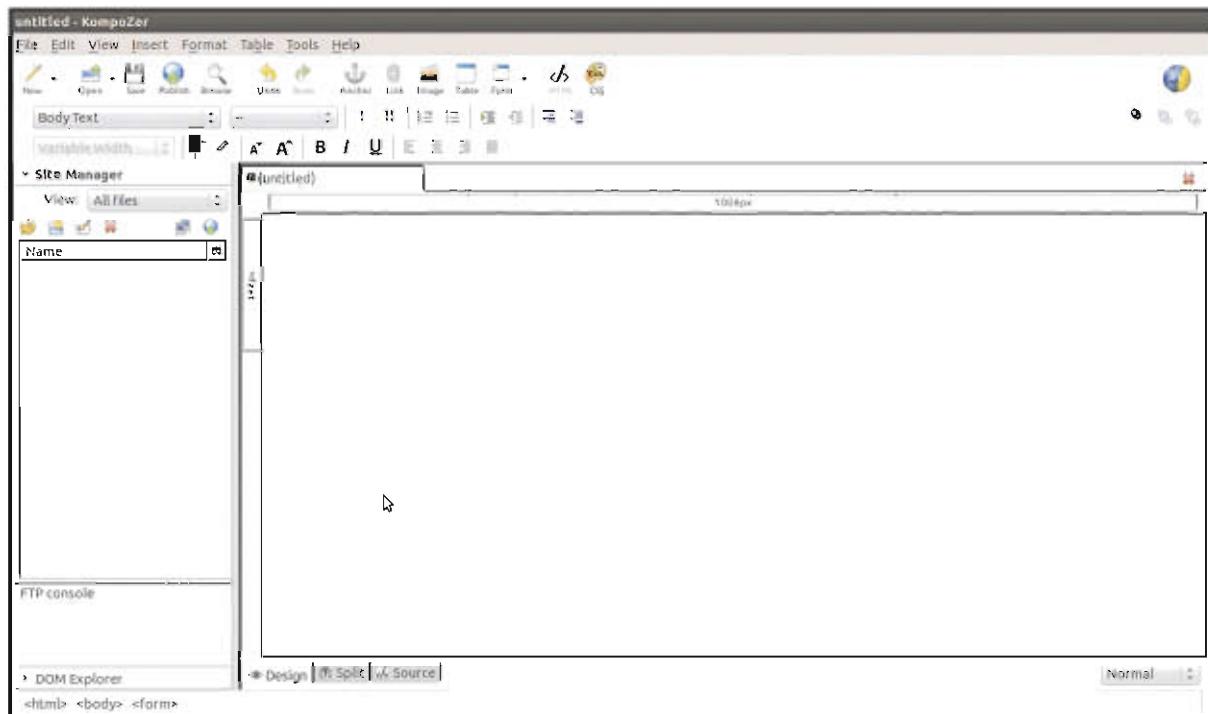


Figure 1.6 : Light blue colored form outline.

- First, we will insert a label for the name field. Click on **Insert → Form → Define label**. Place the cursor in the form where you want your label to appear. Type the text "Name" in the label as shown in figure 1.7. To come out of the label field, click outside the field.

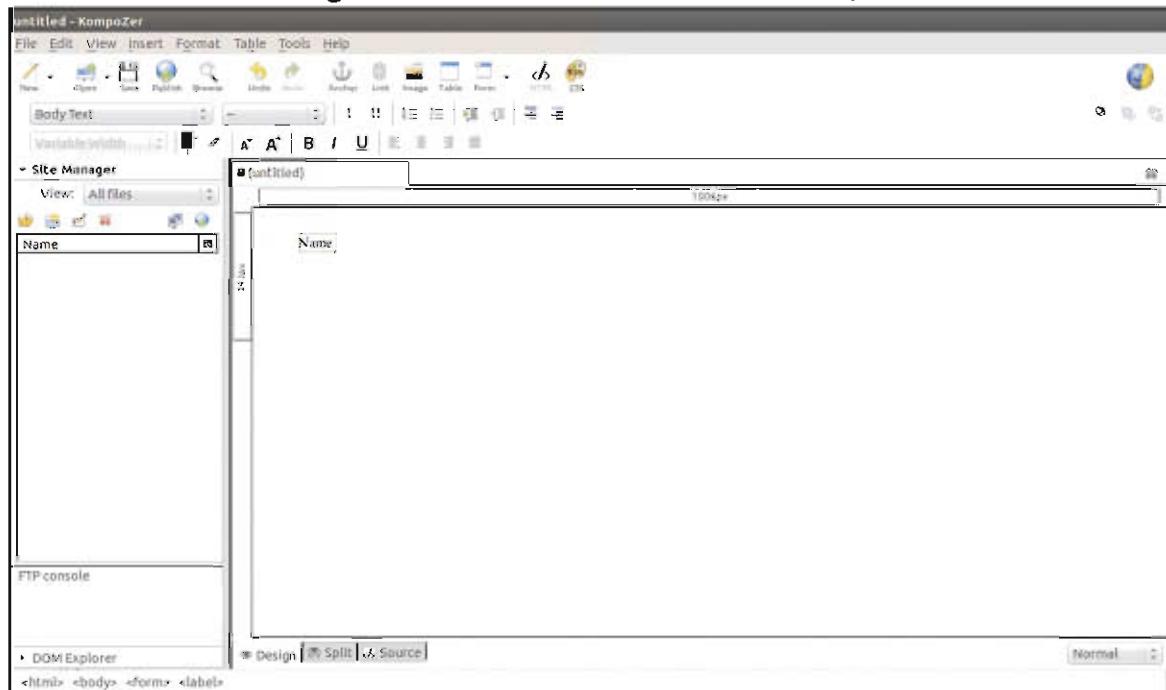


Figure 1.7 : Label field inserted in the form

- To insert an input text field in the form, click **Insert → Form → Form Field**. Figure 1.8 shows the form field properties dialog box. The drop down menu shows various input field type options which we had discussed earlier. Click on more properties, it shows advanced properties related to the field like field size, maximum length.

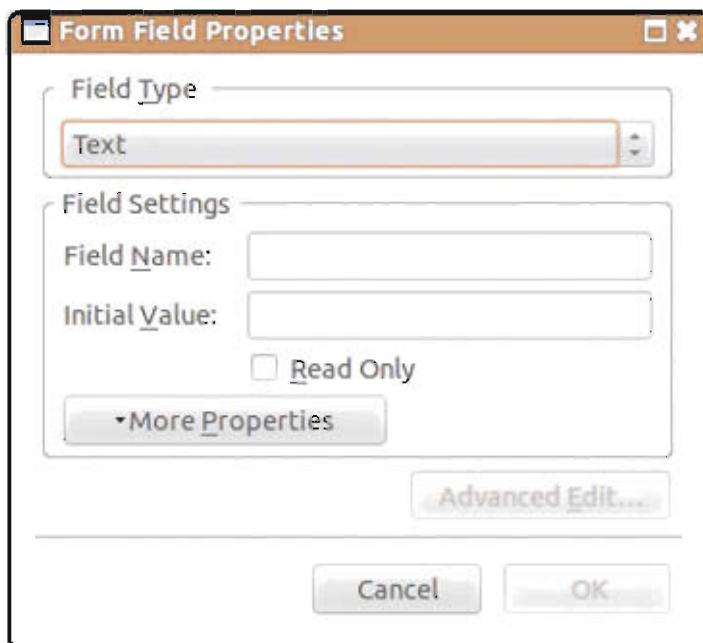


Figure 1.8 : Add form field

- From the drop down menu select Text. Under the Field Settings heading, enter a name in the Field Name text box. In our case we have used name as the field name. Enter some text in the Initial Value field, if you want to show some text before the user actually enters data. Here we have left this field empty. Click on the OK button. Figure 1.9 shows the form after text input field is added.

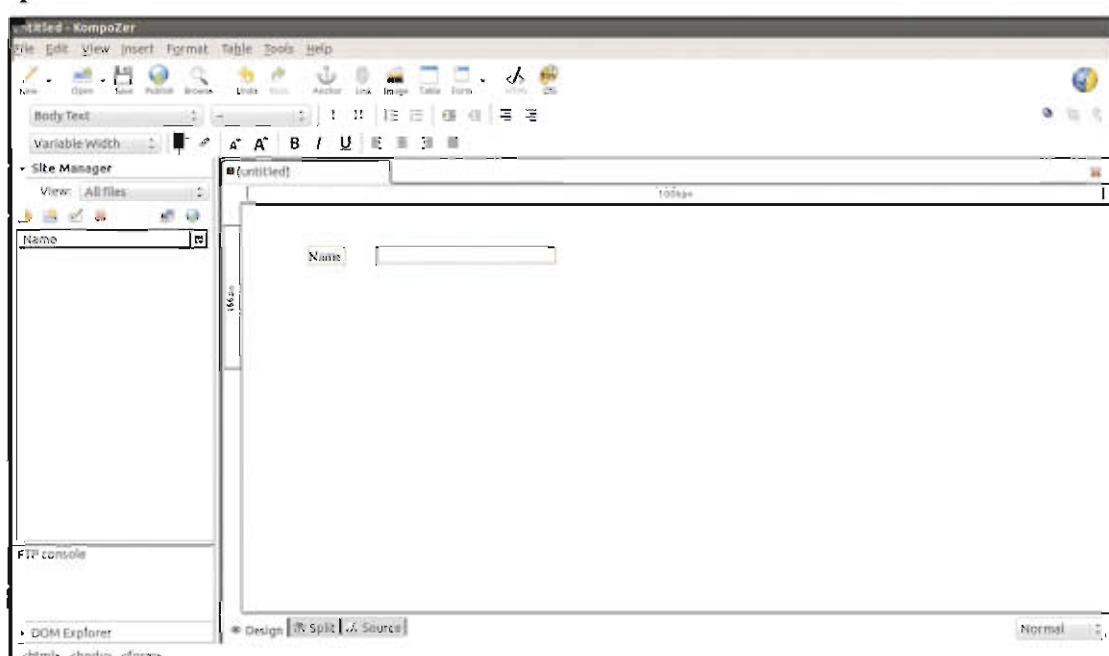


Figure 1.9 : Label and Text field added

- Similarly, we will add another label "E-mail" below the name label field. Add an input text field for E-mail, just like we did for the name field. Here enter text abc@xyz.com in the Initial Value textbox. This will help the user understand the format of E-mail address. Figure 1.10 shows the form with both the text field added.

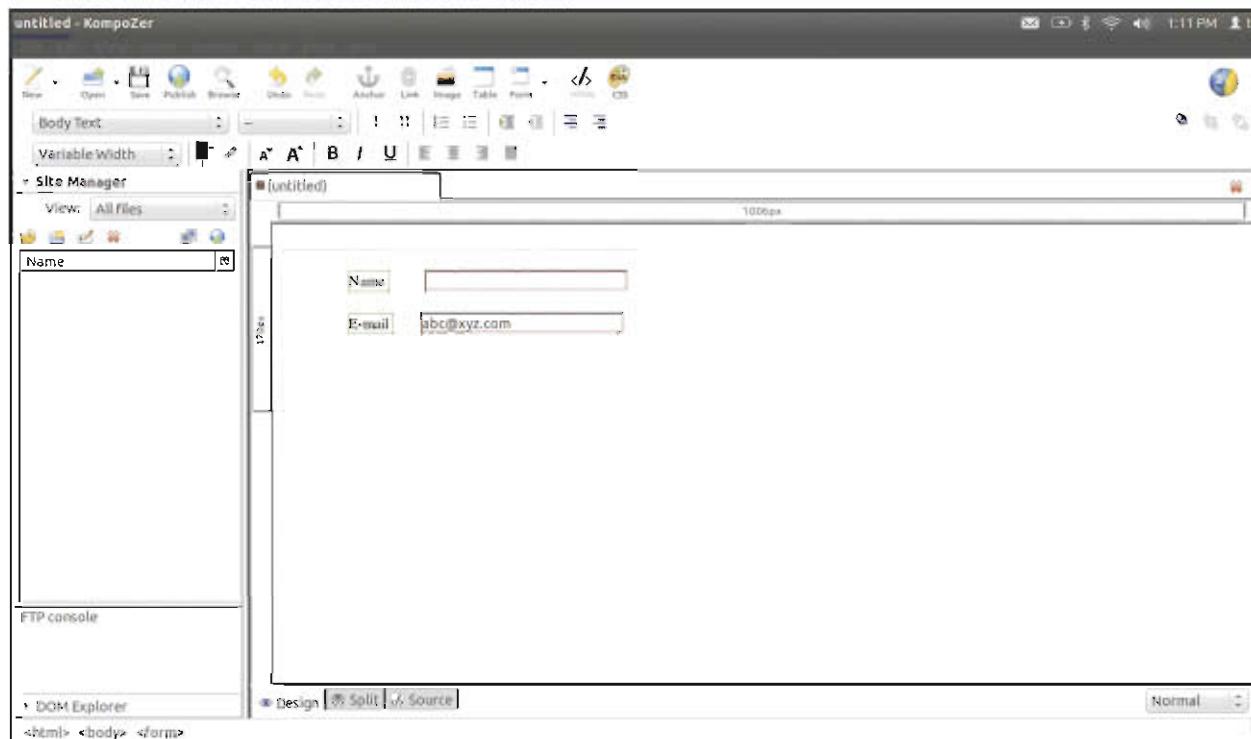


Figure 1.10 : Form with both text fields added

- Finally we will add a submit button in the form. Click **Insert → Form → Form Field**. From the drop-down menu select Submit Button. Type Submit in the both Field Name and Field Value text boxes, and click on the OK button. Figure 1.11 shows the look of form properties dialog box for submit button.

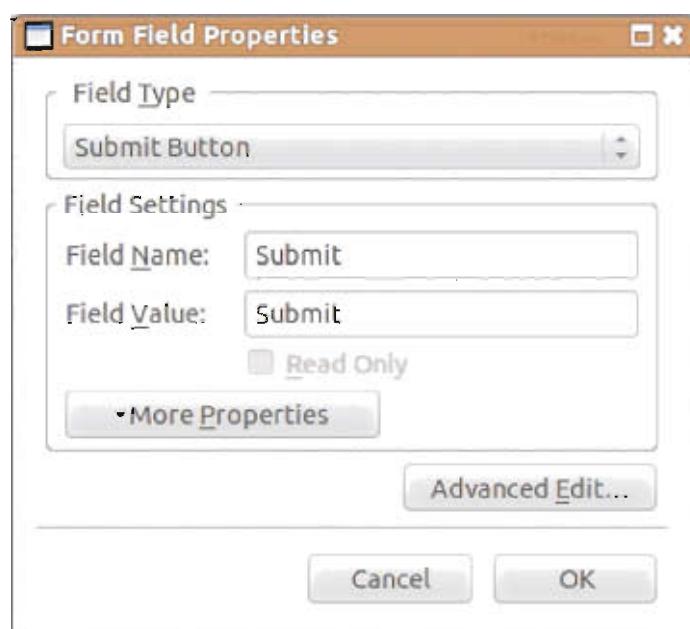


Figure 1.11 : Input field submit button

- The form at present is in the normal view. To have a preview of the form click on Edit mode toolbar and select Preview. Figure 1.12 shows the form in the preview mode.

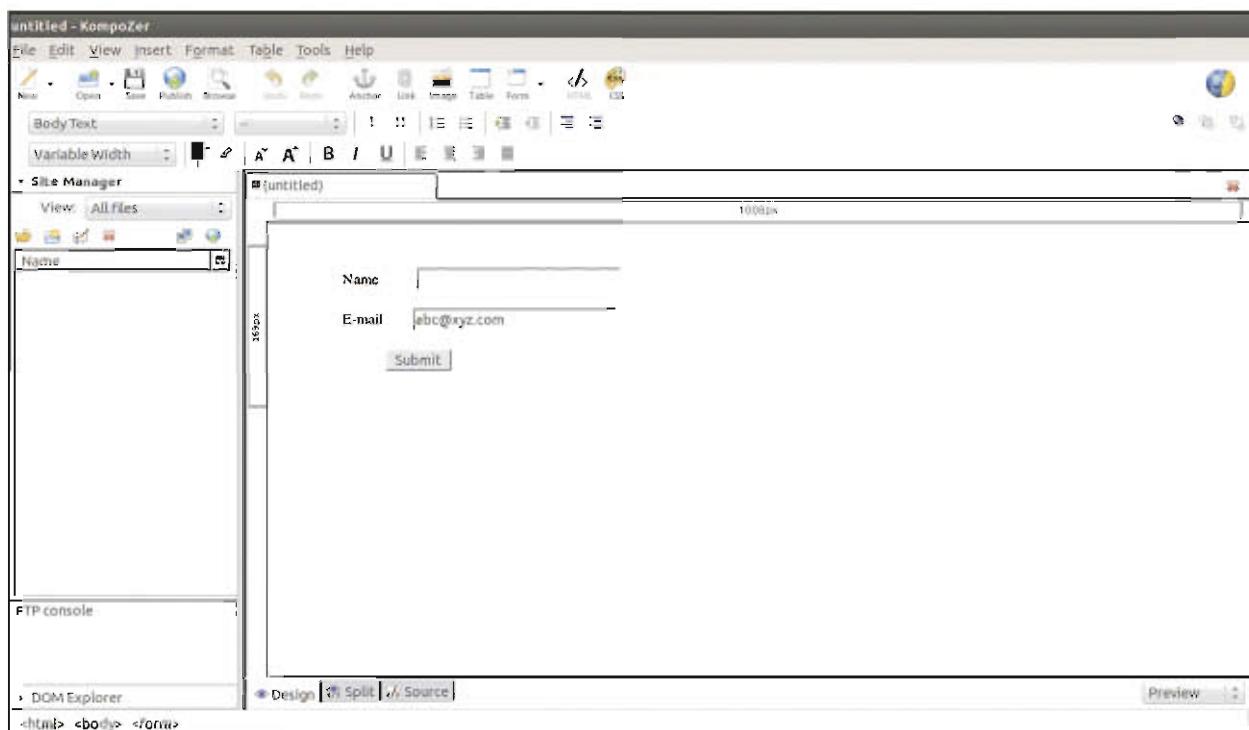


Figure 1.12 : Form as seen in preview mode

Thus, we have created our first form using KompoZer. You can see how KompoZer helps you creating forms within a short span and also relieves you from the tedious job of writing the source code which takes a long time. You can also see the source code of the form just created by clicking on the Source tab. See figure 1.13.

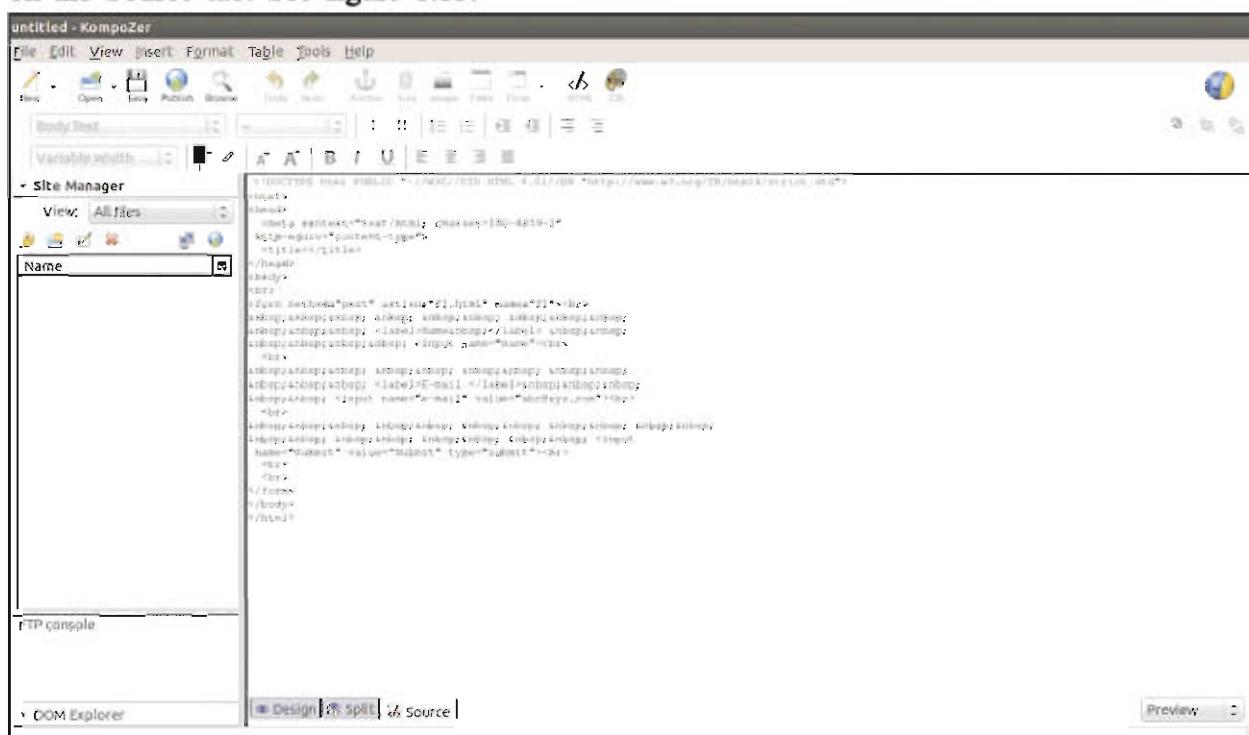


Figure 1.13 : Source code view of form

- Let us save the file which we have created. Click **File → Save**. Alternatively you can also click on the save button  in the composition toolbar. This opens a Page Title dialog box as shown in figure 1.14. We can give a suitable title to the web page here. We have given the name "example1" to the title page. Now click on OK button.

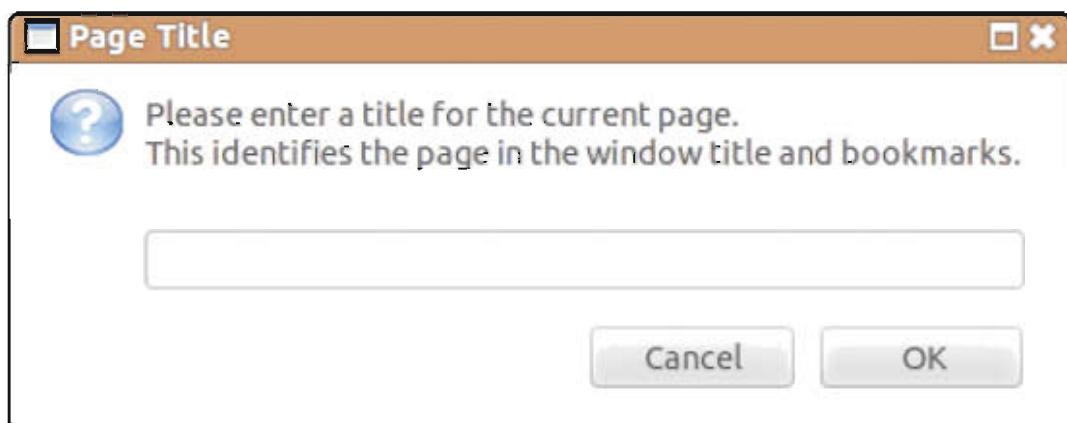


Figure 1.14 : Page title dialog box

The page title will be displayed in the browser windows title bar when viewed in the browser. In case we have created multiple web pages we should give the name of the website as the title page. In this example, since we created a single web page with a form, the title page is named as example1.

- After clicking the OK button, a new dialog box "Save Page As" is opened which prompts to enter a filename and specify the location where you want to save the file as shown in figure 1.15.

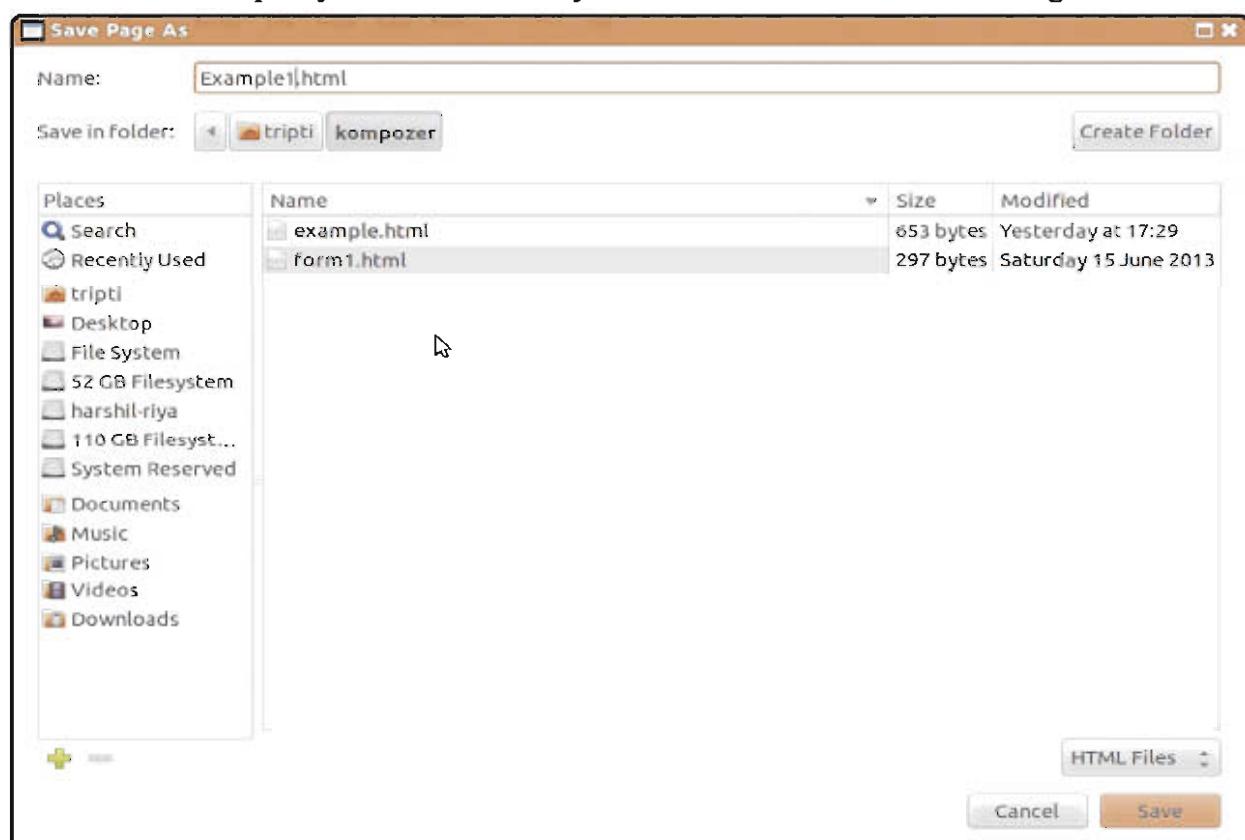


Figure 1.15 : Save Page As Dialog box

Remember to save the file with **html** or **htm** as an extension. Click on Save button. This will take us back to the main window.

Note : If you are creating a website and this page is the home page that will open when you type the website's URL, then save the page with the name index.html.

Having learnt how to create, open, save and create a simple form in KompoZer, let us now create the registration form, which we had created earlier using the HTML tags. Follow the steps given to create the registration form.

- Create a new file.
- From the menu bar, select **Insert → Form → Define Form**. In the Form Properties dialog box enter the details as shown in figure 1.16.

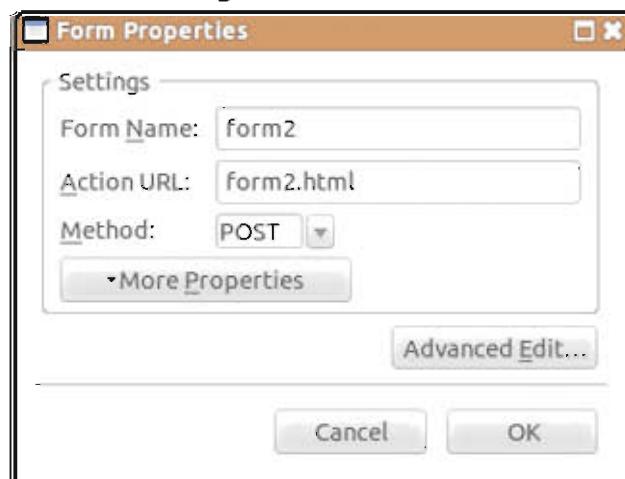


Figure 1.16 : Form Properties dialog box

- Press OK. A form will be displayed showing the light blue colored outline. Press the Enter key to create space in the form.
- To give the heading to the form, select Heading1 from Format Toolbar1. In Format Toolbar2, select centre align icon. Enter the text "Registration Form".
- To insert the label, Click **Form → Define Label**. Type "First Name" in the Field name. Next, to insert input field for the label "First name", click **Form → Form Field**. Select Text in the Field Type menu. Figure 1.17 shows the Form Field Properties dialog box. Observe that we have used "firstname" as the field name. Press OK button.

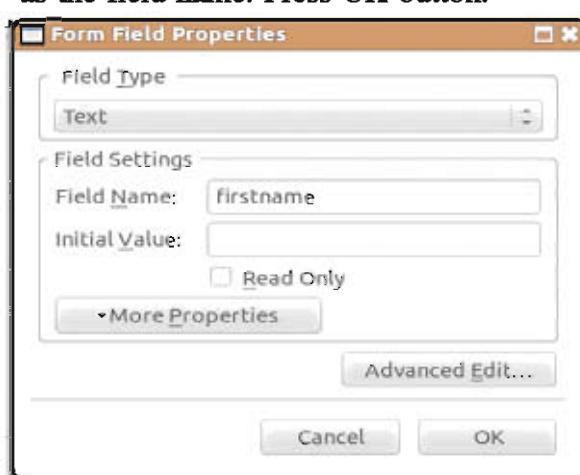


Figure 1.17: Form Field Properties dialog box for Firstname

- Similarly, insert the label "Middle Name" and "Last Name" in the form. The form after adding the fields will look similar to the one shown in figure 1.18.

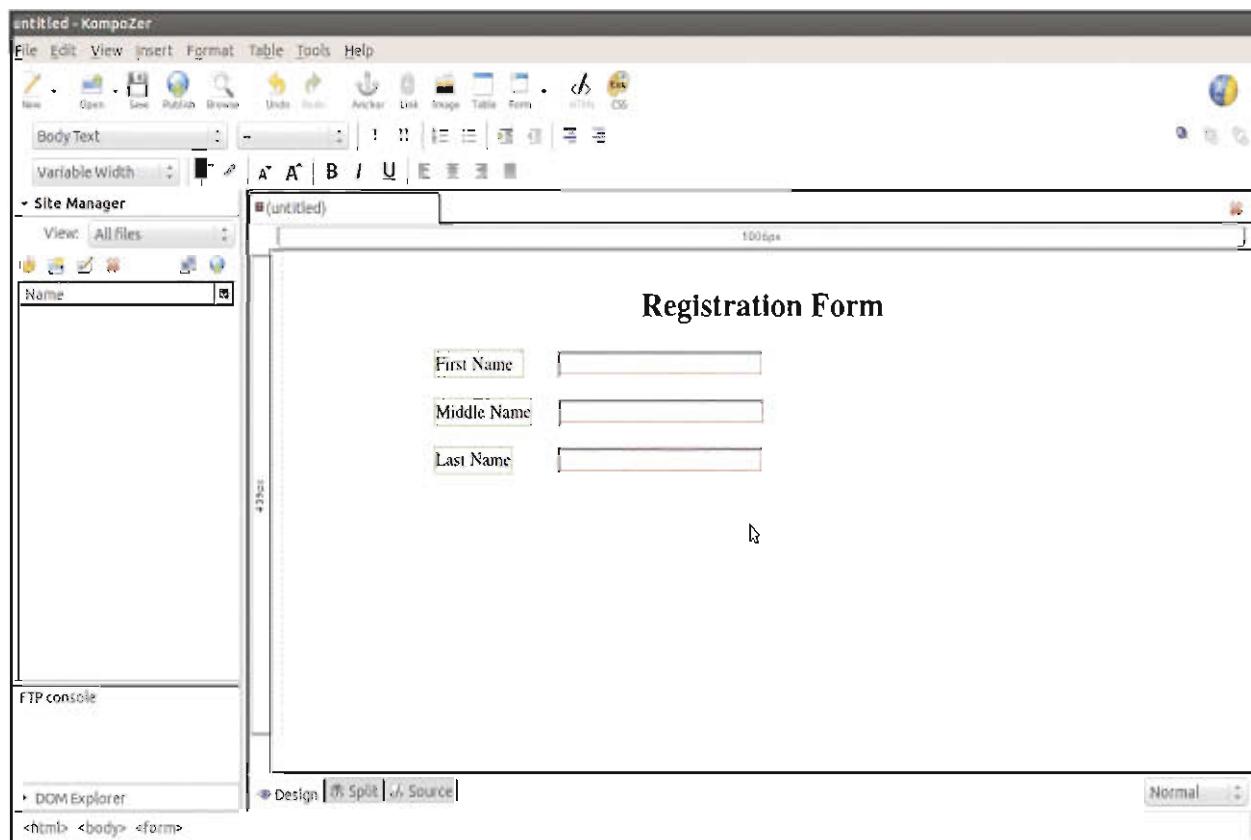


Figure 1.18 : Form displayed after adding the fields

- Now we need to create radio buttons for the Gender field. First, create a label named "Gender".
- For creating a radio button, click **Form** → **Form Field** and select field type as Radio Button from the drop down menu as shown in figure 1.19.

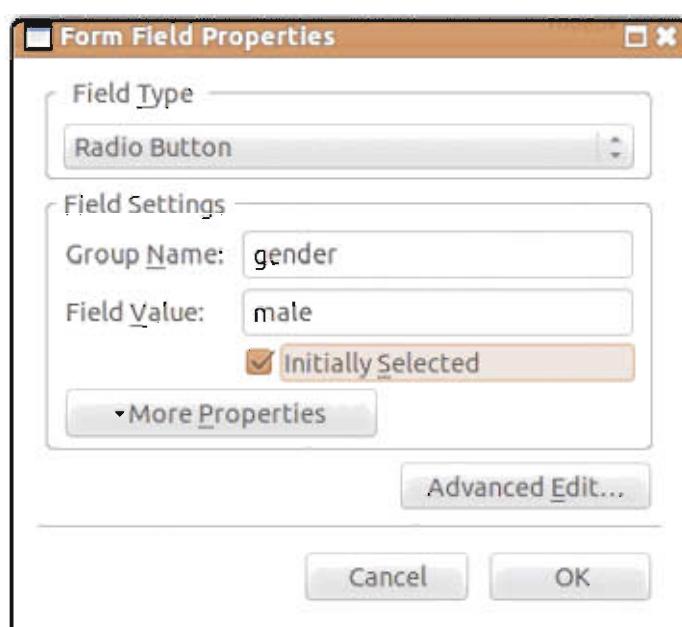


Figure 1.19 : Form field property for Radio Button

Type a name in the Group name box (note that the name should not contain spaces). Here we have entered Group Name as "gender". Similarly in the Field Value text box type "male". If we want the male option of the radio button selected when the form is loaded then, check the box in front of the text "Initially Selected" and click OK button.

- Insert a label with title "Male" near the radio button created.
- Similarly create another radio button named "Female". Remember, when we create the radio buttons within a group, the group name must be the same for all the possible answers. Hence enter the Group Name as "gender". In the Field Value text box type "female". Click OK button.
- Insert a label "Female" near the radio button created. The form after inserting the radio buttons and their labels will look as shown in figure 1.20.

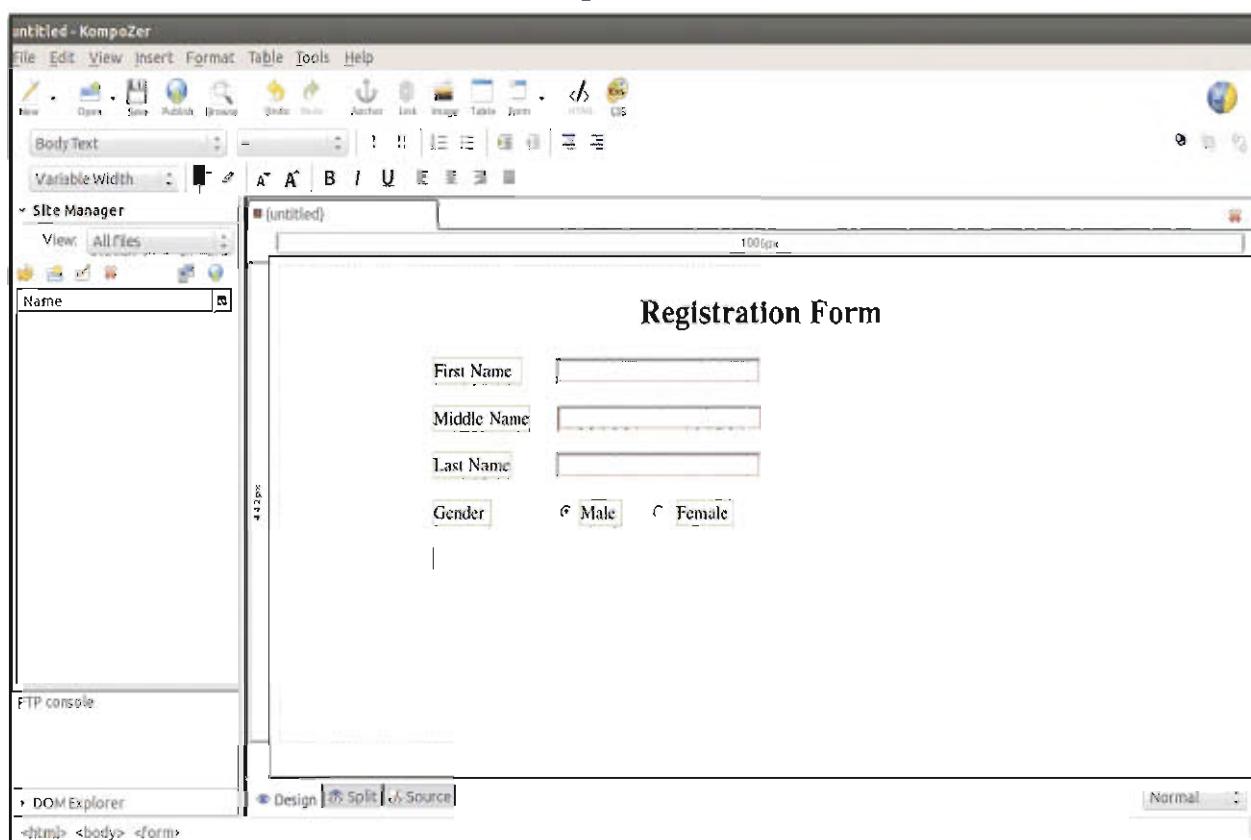


Figure 1.20 : Form displayed after radio buttons are added

- Now we need to add "Hobby" field. As a person can have more than one hobby, hence multiple selections of hobbies are possible. Thus we need to create check box for the hobby field. Create a label for hobby.
- Now, click **Form → Form Field** and from the drop down menu select Check Box field type. Enter name in the Field Name box and a value in the Field Value box as shown in figure 1.21. Check the box in front of "Initially Selected" so as to keep the option checked when the form loads. Click OK button.

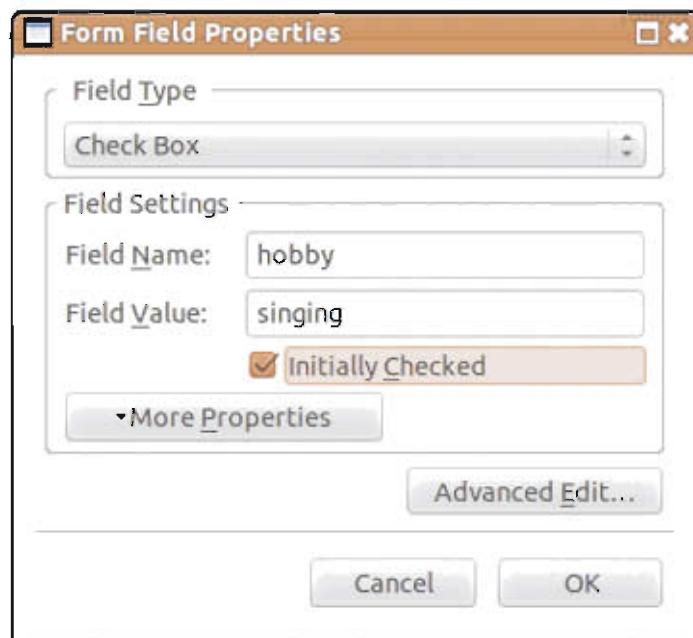


Figure 1.21 : Checkbox field type settings

- Insert a label "Singing" near the checkbox created.
- Similarly create two checkboxes with Field Values as "dancing" and "reading" respectively. Remember, to keep the Field Name same for all the options of checkbox.
- Insert the labels "Dancing" and "Reading" near the checkboxes created. The form after inserting checkboxes and their labels will look as shown in figure 1.22.

The screenshot shows the KompoZer interface with the following details:

- Toolbar:** Includes File, Edit, View, Insert, Format, Table, Tools, Help, and various icons for file operations and styling.
- Site Manager:** Shows a list of files, with 'Name' selected.
- FTP console:** Displays a log message: '2012-07-10 10:06:49 - Connected to localhost'.
- DOM Explorer:** Shows the current document structure: <html> <body> <form>.
- Form Content:** The 'Registration Form' contains the following fields:
 - Text input fields for First Name, Middle Name, and Last Name.
 - Gender selection: Male (radio button) and Female (radio button).
 - Hobby selection: Singing (checkbox), Dancing (checkbox), and Reading (checkbox). The 'Singing' checkbox is checked.
- Bottom Bar:** Includes tabs for Design, Split, and Source, and a Normal view button.

Figure 1.22 : Form display after checkboxes are added

- Next, we need to the "Address" field. As the user enters a large text in the address field, we will keep the field type as textarea. First, create a label named "Address". Now, click **Form → Text Area**. This will open a Text Area Properties dialog box as shown in figure 1.23.



Figure 1.23 : Text Area Properties dialog box

Enter the Field Name. Select the rows and the columns required for the textarea. Here we have kept the rows as 5 and columns as 70. In the Initial Text field enter a suitable text which will be displayed when the form loads. Click OK button.

- Next, we will insert the "City" field. Add a label for the city field. The user will be asked to select the city from the drop down menu. So we need to create the city field using selection list option. Click **Form → Selection List**. This will open a Selection List Properties dialog box as shown in figure 1.24.

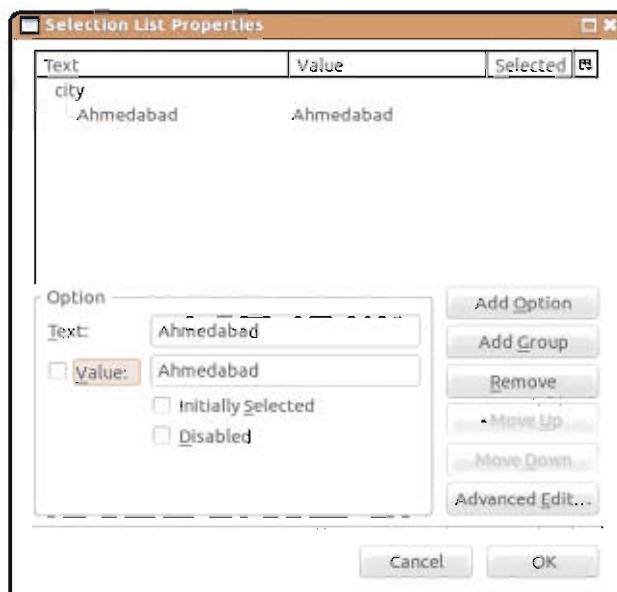


Figure 1.24: Selection List Properties dialog box

Type the name "city" in the List Name box and press Add Option button. Next, type "Ahmedabad" in the Text field. Again press Add option to add the city "Baroda". Likewise, add the city "Rajkot" and "Surat". Remember, to select the option "Initially Selected" when adding the Rajkot city. Click OK button. Figure 1.25 shows the Selection List Properties dialog for the added cities.

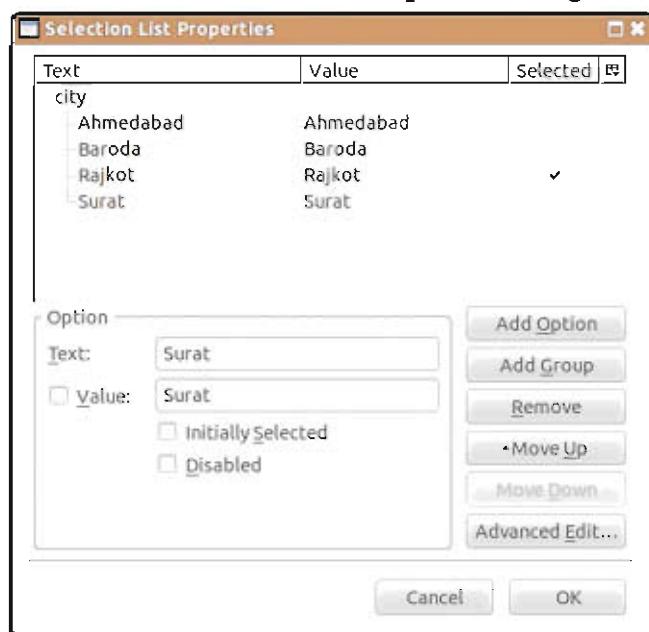


Figure 1.25 : Selection List Properties dialog box for various cities

- Next, we will add the "Submit" button. Click **Form → Form Field**. From the drop down menu select Submit Button. Enter text Submit in Field Name and Field Value text box respectively. Click OK button.
- Similarly we will add the "Reset" button. From the drop down menu select Reset Button. Enter text Reset in Field Name and Field Value text box respectively. Click OK button. The final form in the normal will be as shown in figure 1.26.

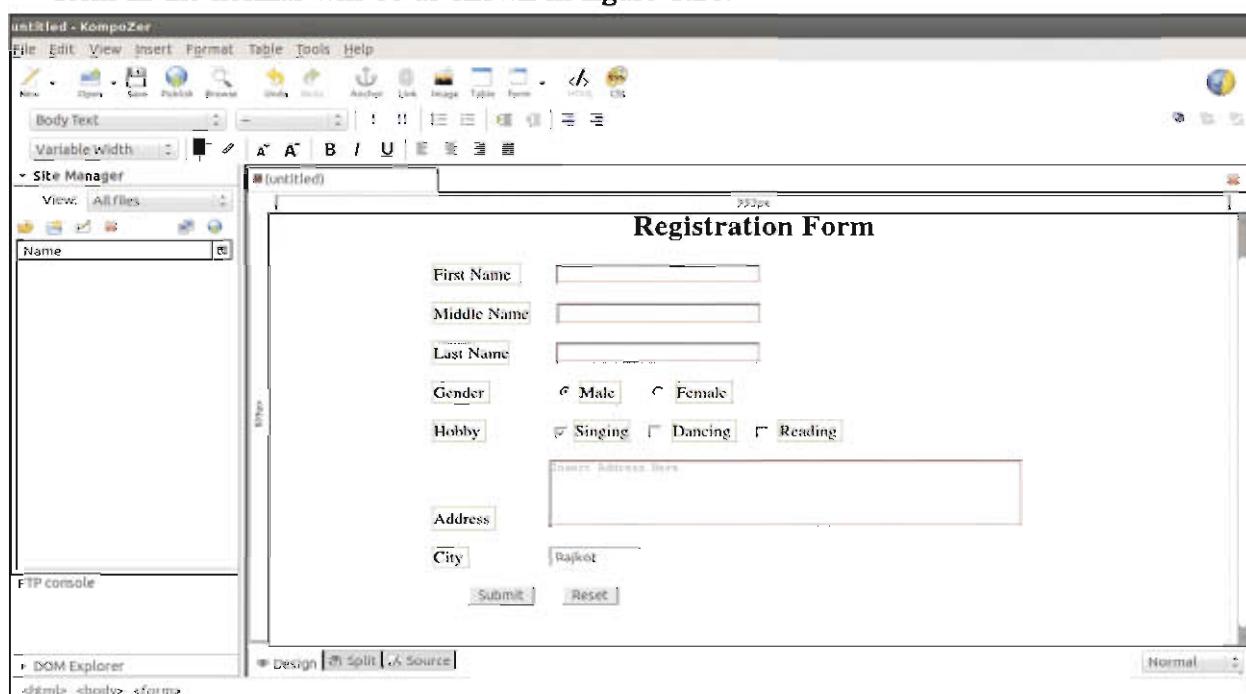


Figure 1.26 : Form displayed in normal view

Save the file with name "example2". Figure 1.27 shows the preview mode of the form.

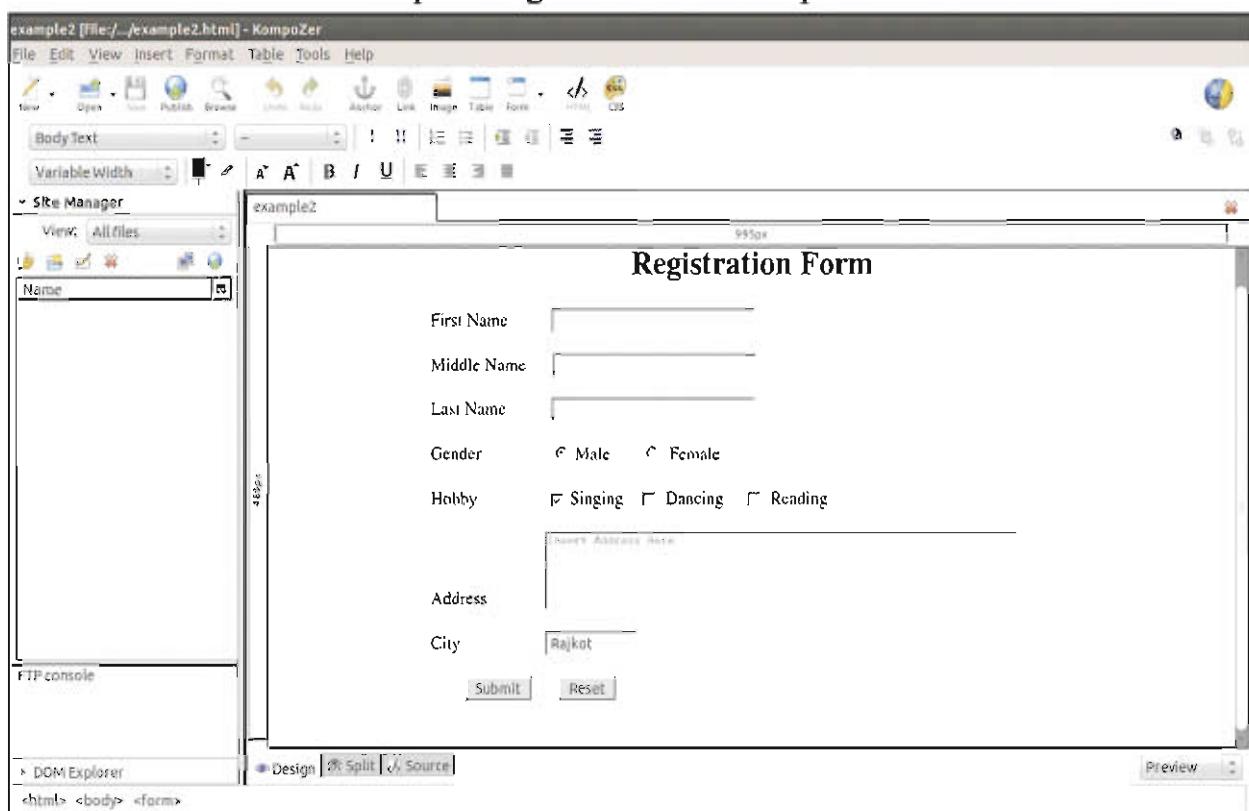


Figure 1.27 : Form displayed in preview mode

Figure 1.28 Shows the registration form as seen in the browser.

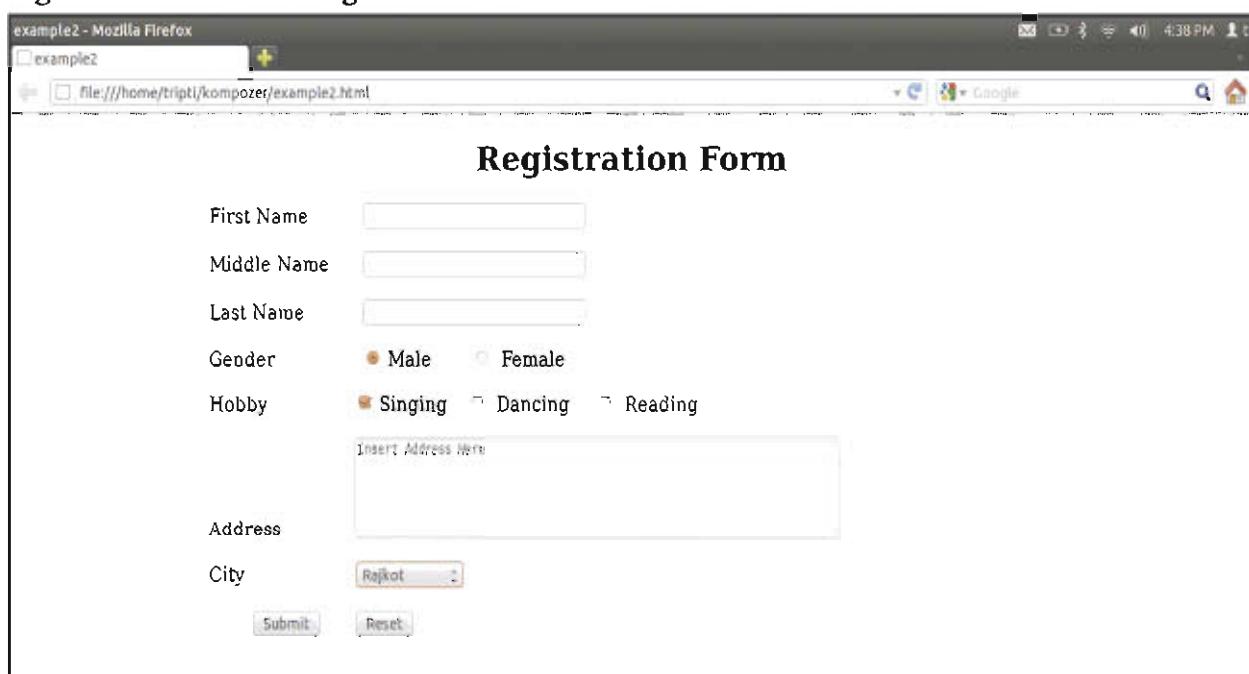


Figure 1.28 : Form displayed in Browser

Observe that the form background is white colored. If you want to give a background color to the form, go to **Format → Page Colors and Background**. This will open a dialog box as shown in figure 1.29.

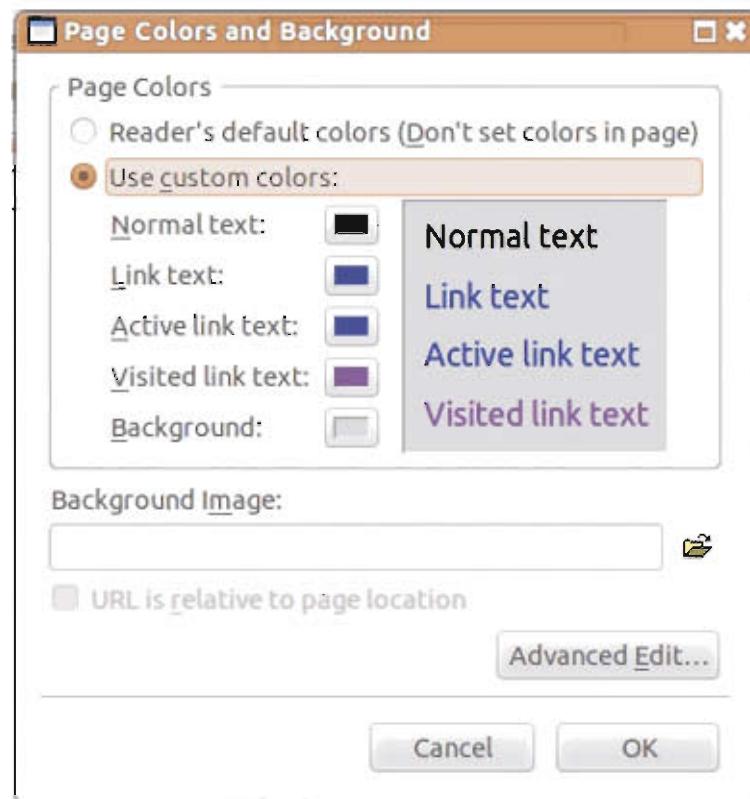


Figure 1.29 : Page Colors and Background dialog box

Select "Use custom colors" option. Click the background option and select the color of your choice from the Block Background Color dialog box as seen in figure 1.30. Click OK button. This will lead you back to the dialog box shown in figure 1.29. Again click OK button.



Figure 1.30 : Background color selection

After selecting the color the form will look as shown in figure 1.31.

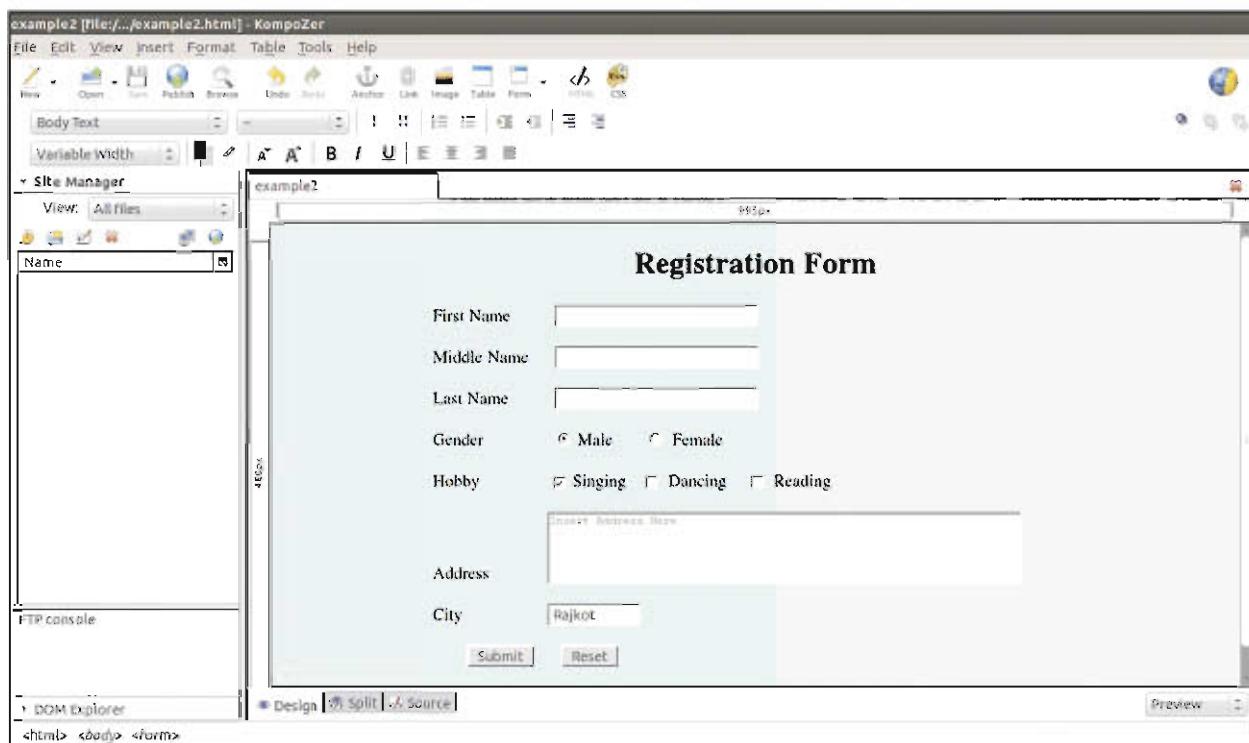


Figure 1.31 : Form displayed in preview mode after adding background color

View the form using the browser and observe the change in the background color.

Summary

Forms are used to accept the data over the web. A form in HTML is a container used to collect different kind of inputs from the user. The users enter the information in the form which can be their personal information, a feedback about a product, a survey or shipping and credit card details. KompoZer is a free open source web development IDE used to create websites. It provides a web page editor which has a simple graphical interface known as WYSIWYG "what you see is what you get". Creating forms using KompoZer is simple and fast.

EXERCISE

1. What is a Form ? List the elements used to create forms in HTML.
2. State the use of Input element in HTML forms. Write about the different attributes of input tag.
3. What is the purpose of textarea element in HTML forms ?
4. Write about select and option element.
5. List the various toolbars seen in the KompoZer interface.
6. Which are the two attributes of form? Explain.

7. Choose the most appropriate option from those given below :

(1) Which of the following is a container used to collect different kinds of inputs from the user.

(a) Form (b) Webpage (c) Text (d) Input

(2) Which of the following element is used to create an HTML form ?

(a) Textarea (b) Form (c) Select and Option (d) Input

(3) Which of the following is the tag used to implement form element ?

(a) <form>... </form> (b) <form>... <form>

(c) </form>... </form> (d) <frm>... </frm>

(4) Which of the following attribute of form is used to specify where to send the form data when the form is submitted ?

(a) method (b) action (c) submit (d) input

(5) Which of the following attribute of form specifies the HTTP method to be used when sending the data ?

(a) submit (b) action (c) method (d) input

(6) Which of the following values are used by method attribute ?

(a) GET and POST (b) GET and SET

(c) GET and PUT (d) SET and POST

(7) Which of the following method allows only a limited amount of information to be sent at a time?

(a) GET (b) POST (c) SET (d) PUT

(8) Which of the following method sends the data as a block through the HTTP transaction ?

(a) GET (b) SET (c) PUT (d) POST

(9) Which of the following attribute of the input element specifies the field that is to be created in the form ?

(a) Input (b) Type (c) Name (d) Value

(10) Which of the following element allows multi-line text input ?

(a) Textarea (b) Input (c) Select and Option (d) Form

(11) Which of the following element is used to create a drop down list or menu in a form ?

(a) Input (b) Textarea (c) Select (d) Form

(13) Which of the following is a free open source web development IDE ?

(a) HTML (b) Kompozer (c) Scite (d) Base

(14) Which of the following stands for "WYSIWYG" ?

(a) When You See Is When You Get (b) What You See Is When You Get
(c) What You See Is What You Get (d) When You See Is What You Get

LABORATORY EXERCISE

1. Create a form for student's personal details.
2. Create a feedback form for the guests who visit your school.
3. You had gone for a vacation with your parents; the tour operator has asked you to give reviews of your trip. Create a form for the same.